ME 350 Lab Exercise 7 Review of Linear Algebra
Winter 2017

Manual Calculations with Linear Algebra

For each of the following exercises, use pencil and paper to obtain the results. Only after you have finished the exercises, check your work with MATLAB.

1. Suppose x, y, and z are column vectors with n elements. Identify the type of result for each of the following expressions. Possible answers are "scalar", "column vector", "row vector", "matrix", and "illegal".

   (a) $x^T y$

   (b) $(z^T y)z$

   (c) $(x^T y)^T$

   (d) xy^T

   (e) $(x + y)^T z$

2. Given $x = [1, 2, 3]$ and $y = [3, 2, 1]^T$ evaluate the following expressions, or state "illegal" if those operations are not allowed by the normal rules of linear algebra

   (a) $x/4$
   (b) $x + y$
   (c) xy
   (d) yx
   (e) $\|x\|_2 - \|y\|_2$

3. Given $A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 0 & 0 \end{bmatrix}$ and $x = \begin{bmatrix} 2 \\ 3 \\ 3 \end{bmatrix}$.

   (a) Compute Ax
   (b) Compute $x^T A$

Linear Algebra with MATLAB

Download `vectorNormShell.m` and rename it as `vectorNorm.m`

1. Complete the code in each of the three subfunctions, `L1norm`, `L2norm` and `Linfnorm`. Remember the definitions of the p-norms

$$\|x\|_2 = \left(x_1^2 + x_2^2 + \cdots + x_n^2\right)^{1/2} = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}. \tag{1}$$

$$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n| = \sum_{i=1}^{n} |x_i|, \tag{2}$$

$$\|x\|_\infty = \max\left(|x_1|, |x_2|, \ldots, |x_n|\right) = \max_i\left(|x_i|\right). \tag{3}$$

2. Test your code with the following statements

```
x = rand(5000,1);
e1 = vectorNorm(x,1) - norm(x,1)
e2 = vectorNorm(x) - norm(x)
einf = vectorNorm(x,Inf) - norm(x,Inf)
```

Minimum Residual for Least Squares Fit

Let x_i and y_i denote a given data pair in a set of n ordered pairs (x, y). A least squares line fit gives an estimate of y_i as

$$y_{i,\mathrm{fit}} = c_1 x_i + c_2 \tag{4}$$

where c_1 is the slope and c_2 is the intercept of the fit. The least squares algorithm guarantees that

$$\rho = \|r\|_2 \tag{5}$$

is a minimum for all possible choices of c_1 and c_2, where

$$r = y_{\mathrm{fit}} - y \tag{6}$$

is the residual vector of the least squares fit.

The built-in `polyfit` function computes the least squares fit of a polynomial of any order to a given data set. The generic formula is

```
c = polyfit(xdata,ydata,n)
```

where *`xdata`* and *`ydata`* are the given data vectors, and *`n`* is the degree of the fit polynomial. For a line fit, *`n`*= 1. The output of `polyfit` is a vector of coefficients in decending powers of x. In general, for a degree n polynomial

$$y_{\mathrm{fit}}(x) = c_1 x^n + c_2 x^{n-1} + \cdots + c_n x + c_{n+1} \tag{7}$$

and for a line

$$y_{\mathrm{fit}}(x) = c_1 x + c_2 \tag{8}$$

which is the same as Equation (4), except that Equation (4) is written for an individual x_i and Equation (**??**) is written for any x.

Given a vector of polynomial coefficients, `c`, output from the `polyfit` function, the `polyval` function evaluates the polynomial stored in `c`

```
yfit = polyval(c,xfit)
```

where `xfit` is a scalar or vector of x values at which the fit is to be evaluated.

The `fitLine` function in Listing 1 shows how to compute and plot a least squares line fit to (x, y) data pairs.

Exercises

1. Run the `fitLine` function and verify the the least squares line fit at least looks plausible

2. Modify the `fitLine` function by removing the comment symbols from these two lines at the end of the `fitLine`:

   ```
   % rfit = %% INSERT YOUR CODE HERE
   % fprintf('Least squares fit has residual   = %12.3e\n',rfit);
   ```

   Replace the `%% INSERT YOUR CODE HERE` with code to evaluate

   $$\rho = \|r\|_2 = \|y_{\text{fit}}(x) - y\|_2$$

   where $y_{\text{fit}}(x)$ is the vector of fit function values at the original x data, and y is the original data. In other words, the elements of $y_{\text{fit}}(x) - y$ are $y_{\text{fit}}(x_i) - y_i$..

3. Download the `randLineFitShell.m` file and save it as `randLineFit.m`. Complete the missing lines of code.

   (a) In line 36

   ```
   r = %% INSERT code to compute L2 norm of residual
   ```

   the missing code is similar to the code you inserted into the `fitLine` function

   (b) In line 43

   ```
   %   if %% INSERT test to determine whether current residual is minimum
   ```

   you test the value of `r` computed in line 36 against `minr` to determine whether to replace `minr` with the new `r`. When this `if` statement is completed, remove the comment characters from the start of lines 43 through 50.

   The goal of `randLineFit` is to demonstrate that the least squares fit minimizes the residual of the fit function.

```
function fitLine
% fitLine  Demonstrate least squares fit of a line to data

% -- Store the data set
x = 1:8;
y = [4.41  6.82  7.07  12.02  13.97 20.77 20.87  25.17];

% -- Polyfit is built-in command to compute polynomial curve fit
c = polyfit(x,y,1);

% -- Generate data to evalute the fit over the range of the data
xfit = [min(x) max(x)];
yfit = polyval(c,xfit);

% -- Plot the data and the curve fit
plot(x,y,’o’,xfit,yfit,’r--’);
legend(’Data’,’Line fit’,’Location’,’northwest’)

% -- Evaluate the residual of the fit at the data points
% rfit = %% INSERT YOUR CODE HERE
% fprintf(’Least squares fit has residual   = %12.3e\n’,rfit);

end
```

Listing 1: The `fitLine` function demonstrates a least squares fit of a line to data.

```
function randLineFit(ns,slope,intercept)
% randLineFit  Randomly select slope and intercept to fit a line to data

if nargin<1, ns=200;            end
if nargin<2, slope=1;           end
if nargin<3, intercept = 0.05;  end

% -- Plot data set
x = 1:8;
y = [4.41  6.82  7.07  12.02  13.97 20.77 20.87  25.17];
gray = [0.7 0.7 0.7];
plot(x,y,'.','MarkerSize',18,'Color',gray)
axis([0 10 0 30]);
hold('on')

% -- Generate a list of random slope and intercept values based on
%    user's guess at slope and intercept
slopeRand     = slope + 2*randn(ns,1);
interceptRand = intercept + 2*randn(ns,1);

% -- Set first guess at first random slope and intercept
ybest = zeros(size(y)); %  Will be replaced by better set of y values
rsave = zeros(ns,1);    %  vector to store residuals
kbest = 0;              %  Index of best random slope & intercept
nreplaced = 0;          %  Number of times best guess is replaced
minr = 1e6;             %  Guarantee that 1st guess will be minimum residual

for i=1:ns

  % -- Evaluate next guess line at given data points
  ytest_points = slopeRand(i)*x + interceptRand(i);

  % -- Use L2 norm to measure difference between y values from this
  %    guess and the y values

  r = %% INSERT code to compute L2 norm of residual

  rsave(i) = r;

  % -- Save the line with smallest norm and update best guess line
  %    with the new line if it is has the current smallest norm

%   if %% INSERT test to determine whether current residual is minimum
%
%     minr = r;
%     kbest = i;
%     nreplaced = nreplaced + 1;
%     ybest = ytest_points;
%
%   end

end

fprintf('\nAfter %d tries and %d replacements\n',ns,nreplaced);
fprintf('the best random fit has residual = %12.3e\n',rsave(kbest));

%%
%%  More plotting code here

end
```

Listing 2: The `randLineFitShell` function is incomplete. When finished the code provides empirical justification for the idea that the least squares fit obtains the minimum residual for the fit function.